Advanced Automata Theory (Summer 2017) Problem Set #1 (Due 03.05.2017)

Problem 1.1. A word $x \in \Sigma^*$ is called a *prefix* of a word $y \in \Sigma^*$ if there is a word $z \in \Sigma^*$ such that $y = xz$. Let L be a language. Define

$$\mathit{prefix}(L) = \{w \in \Sigma^* \mid \exists x \in L : w \text{ is a prefix of } x\}$$

Show that if L is regular then $\mathit{prefix}(L)$ is regular by modifying a DFA for L to accept $\mathit{prefix}(L)$.

Problem 1.2. We mentioned in class that any DFA for the language

$$L_n = \{w \in \{0,1\}^* \mid \text{the } n\text{th symbol from the right is a } 0\}$$

has 2^n states. Prove this. [Hint: If a DFA for L_n has fewer than 2^n states, then there are two strings of length n such that the DFA comes to the same state after reading them (why?). Show how you can use this to prove that the DFA does not accept L_n.]

Problem 1.3. Show that every finite set is regular. Are regular languages closed under infinite union? That is, if L_1, L_2, ... are infinitely many regular languages, must $\cup_{i\geq 1} L_i$ also be regular?

Problem 1.4. A *unary* language is a language over a one-letter alphabet. For example, if $\Sigma = \{0\}$, a unary language is a subset of 0^*. Give an example of a unary language that is not regular. Prove that your language is not regular.

Problem 1.5. For languages L_1 and L_2, define the *shuffle* of L_1 and L_2 to be the language

$$\{w \mid w = a_1 b_1 \dots a_k b_k, \text{ where } a_1 \dots a_k \in L_1 \text{ and } b_1 \dots b_k \in L_2, \text{ each } a_i, b_i \in \Sigma^*\}$$

Show that the class of regular languages is closed under shuffle.

Problem 1.6. [*] Suppose $L_1 \subseteq L_2$ be regular languages such that $L_2 \setminus L_1$ is infinite. Show that there is a regular language L such that $L_1 \subseteq L \subseteq L_2$ and $L_2 \setminus L$ and $L \setminus L_1$ are both infinite. ("*"ed problems a bit harder than others.)

Problem 1.7. Give an algorithm to check that an NFA only accepts words of odd length.

Problem 1.8. Give an algorithm to check if the language of an NFA is infinite.

Problem 1.9. Consider an alternate definition of quotient, defined as follows for $L_1, L_2 \subseteq \Sigma^*$:

$$L_1//L_2 = \{x \in \Sigma^* \mid xy \in L_1 \text{ for all } y \in L_2\}$$

If R is regular and L is any language, must $R//L$ be regular?